
Test Driven Development

Dipping your toe into tests first

BrumPHP - 27th June 2019

Mark Bradley
@braddle
@testingAllThe

What?

What is TDD?

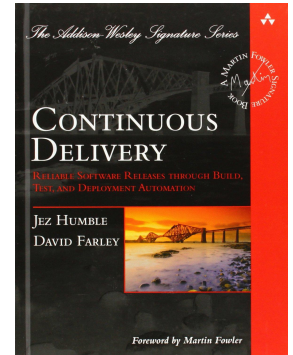
Writing test before code

Using test to design the application

Not just unit tests

Why?

Why I do use TDD?



Boring

Tests are hard to write

Easy to start skipping test

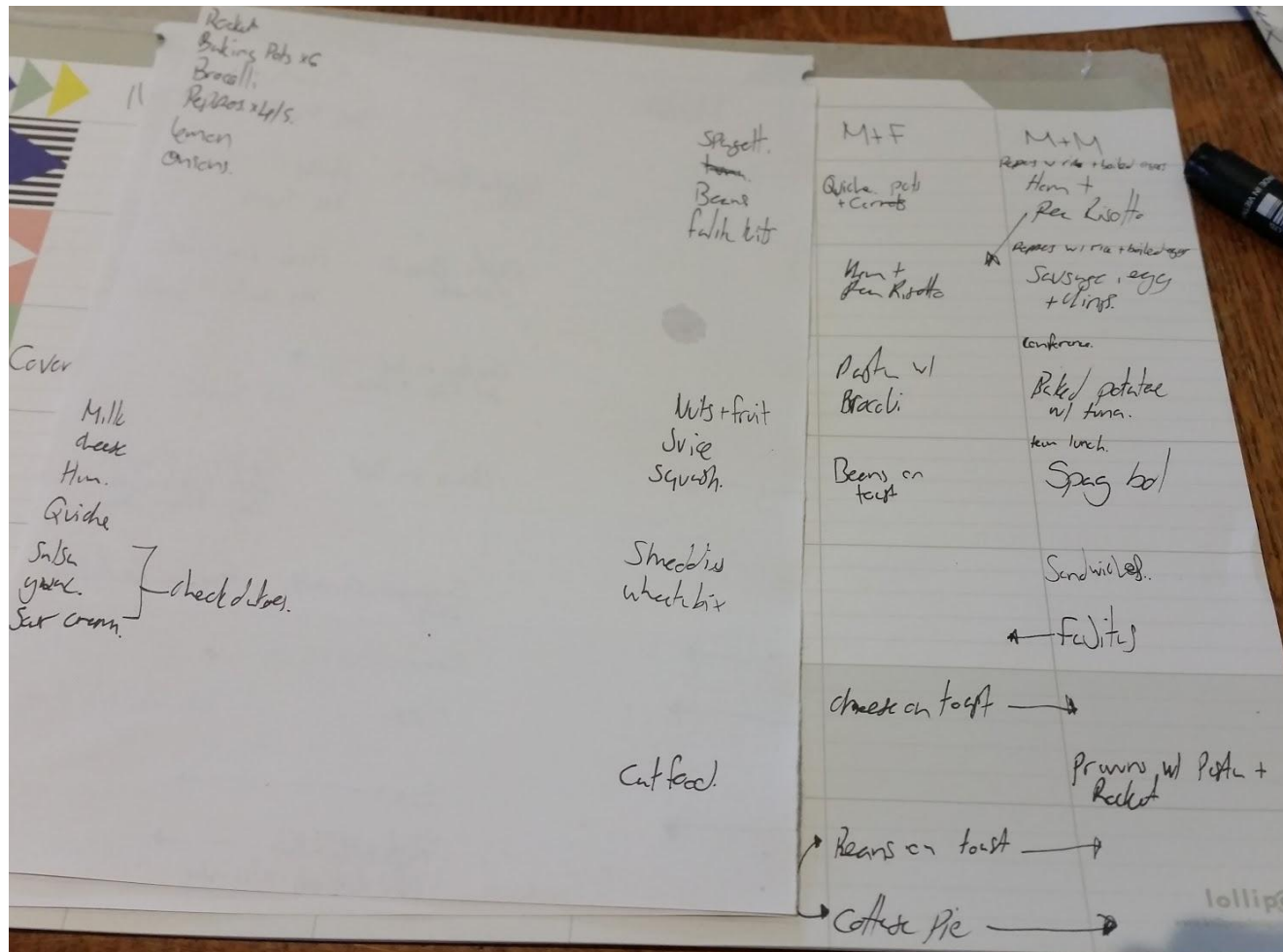


Fun!

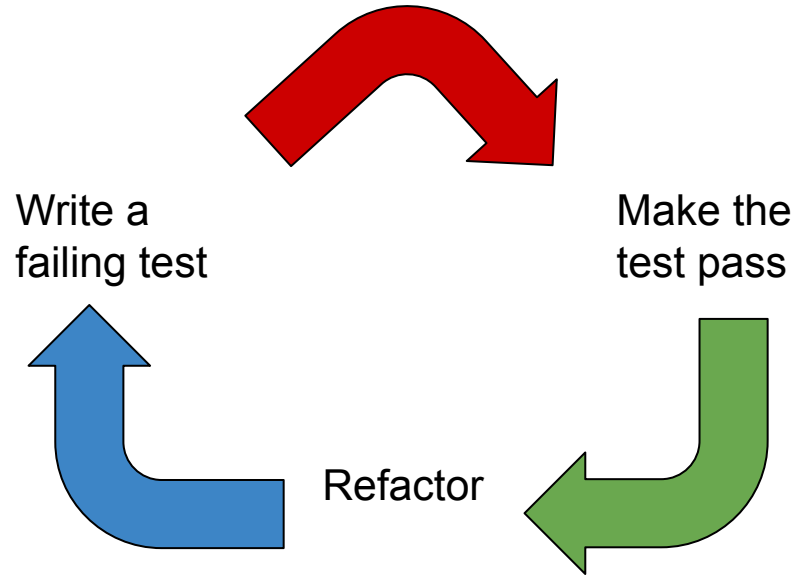
Easy to write

—

You ~~don't~~ can't skip tests

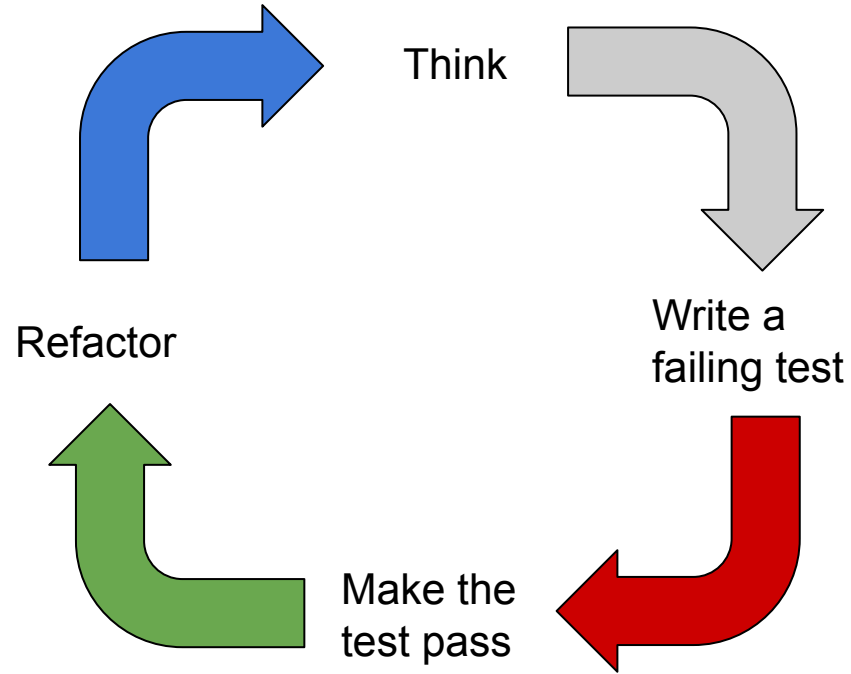


TDD Cycle



Does it slow down development?

TDD Cycle

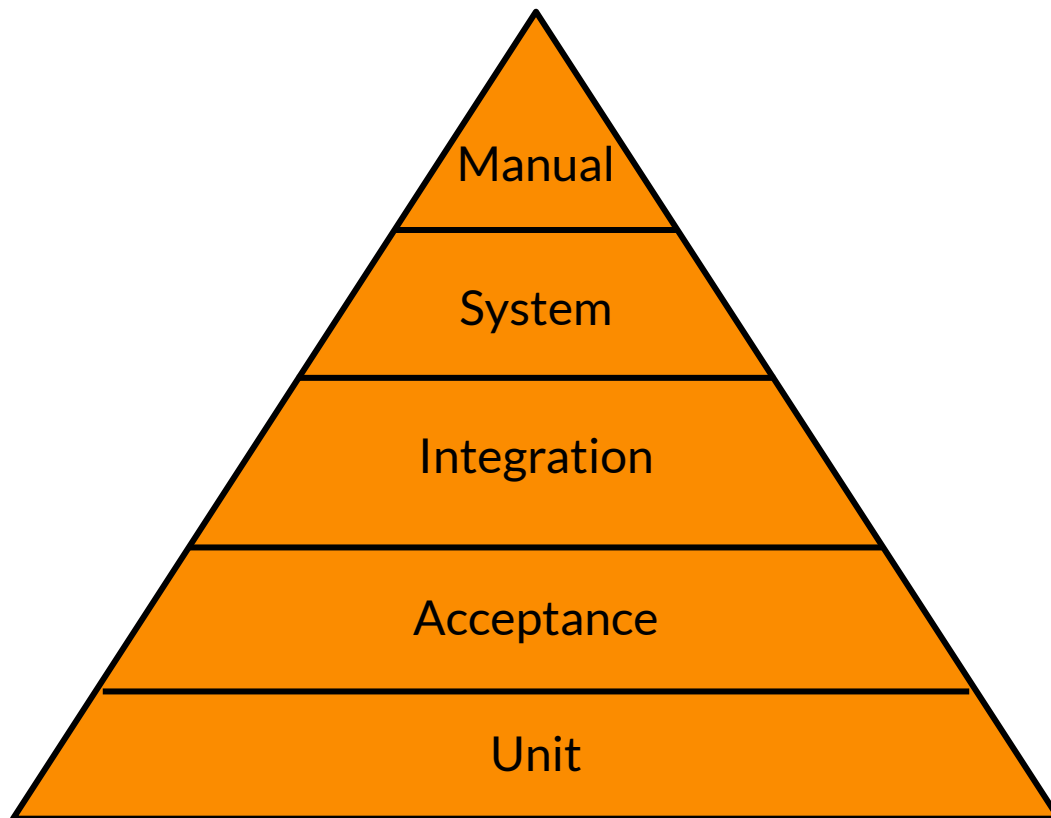


3 Rules of TDD

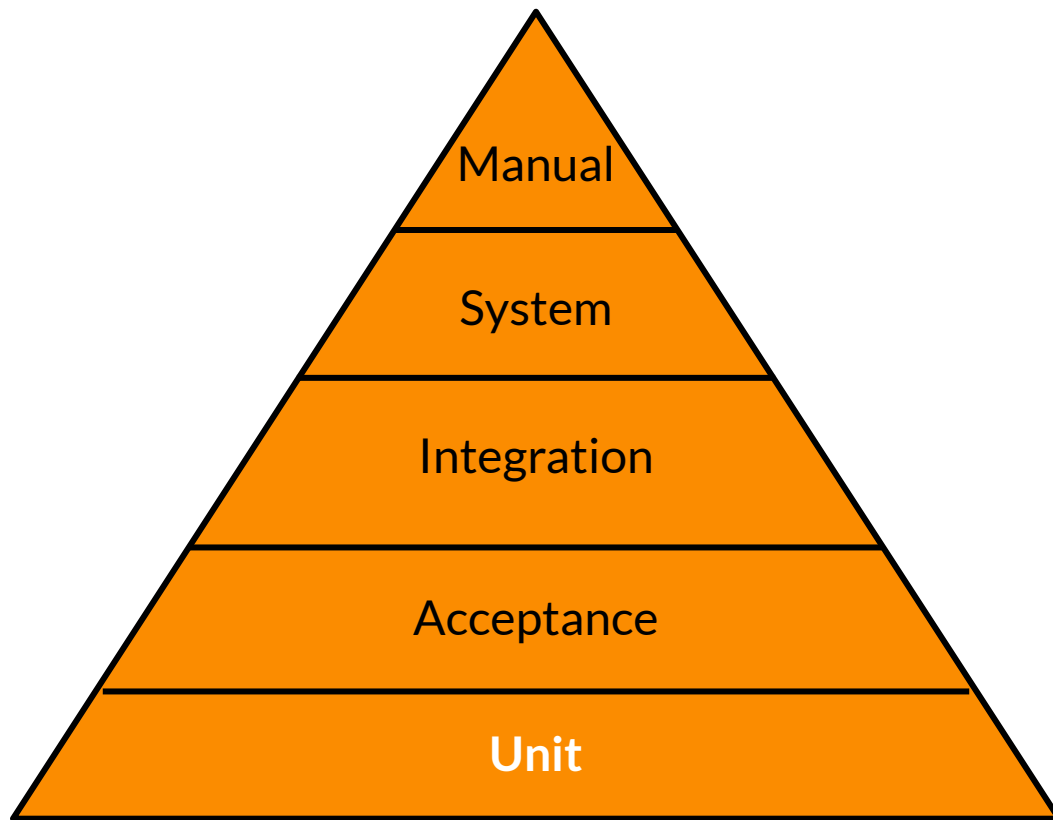
1. Write no production code until you write a failing test.
2. Do not write more of a test than to make the test fail (compilation error are test failure).
3. Do not write more production code than is enough to make the test pass.

- Uncle Bob

Testing Pyramid



Unit Tests



Unit Tests

Targeted

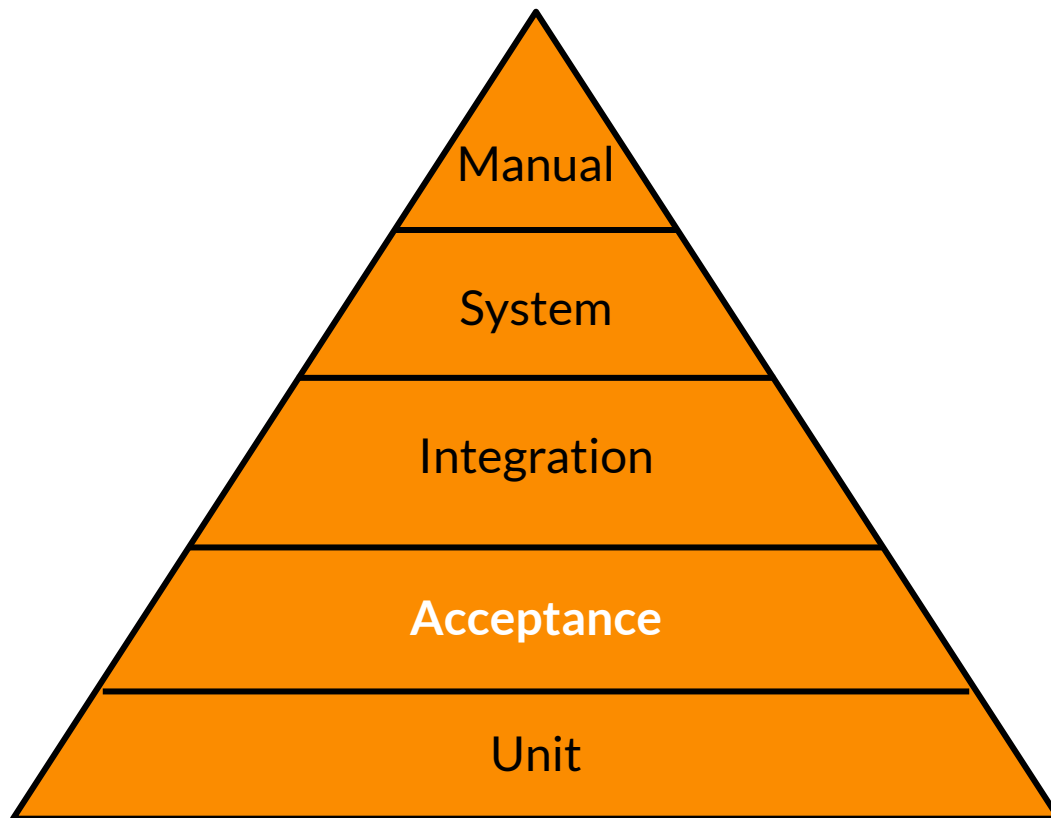
Isolated

Repeatable & predictable

Fast

100% Code Coverage

Acceptance Tests



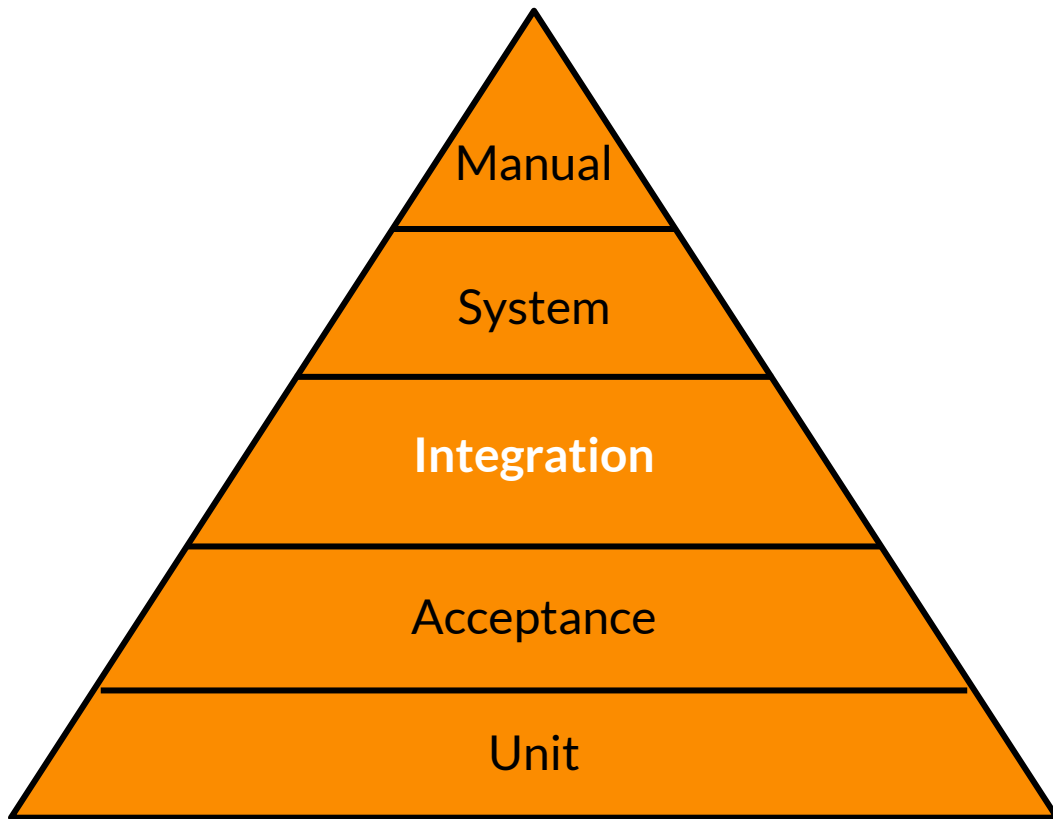
Acceptance Tests

Ensure Business requirement are met

Define by Product Owners

50% Code Coverage

Integration Tests



Integration Tests

Tests a small number of units work together

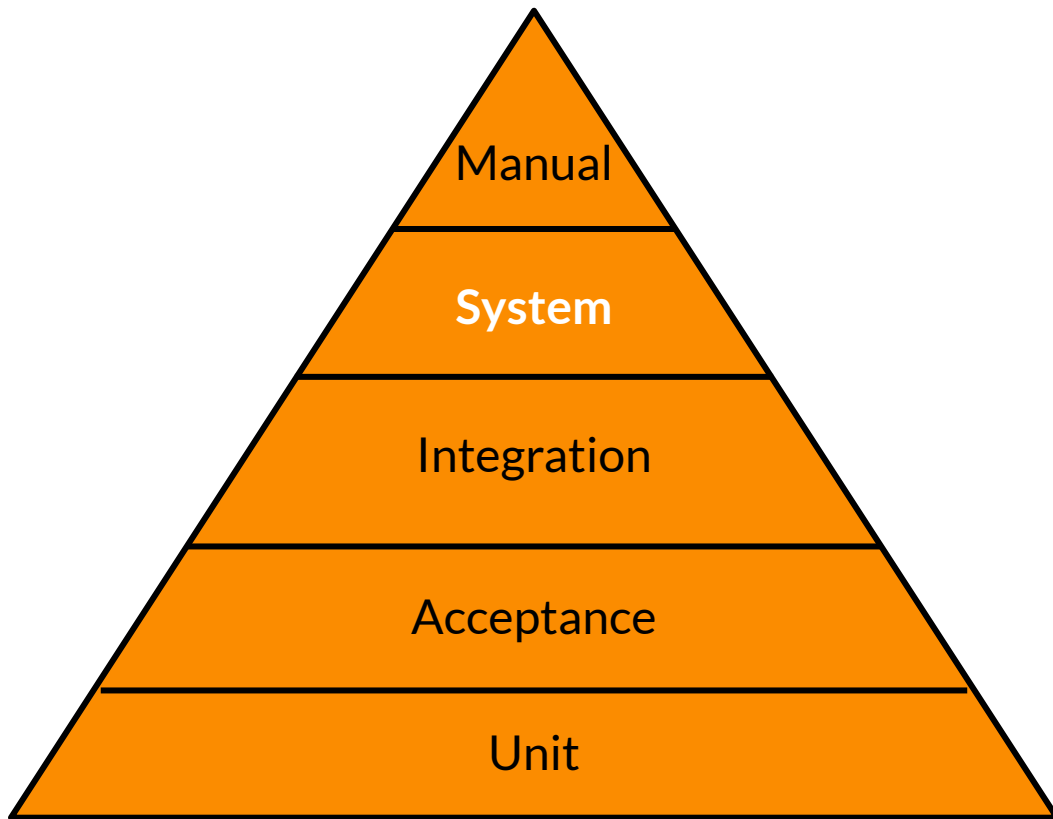
Test the integration points of you application

10% Code Coverage



twitter.com/ThePracticalDev

System || End to End Tests



System (End to End) Tests

Flows through you application, usually a few core journeys

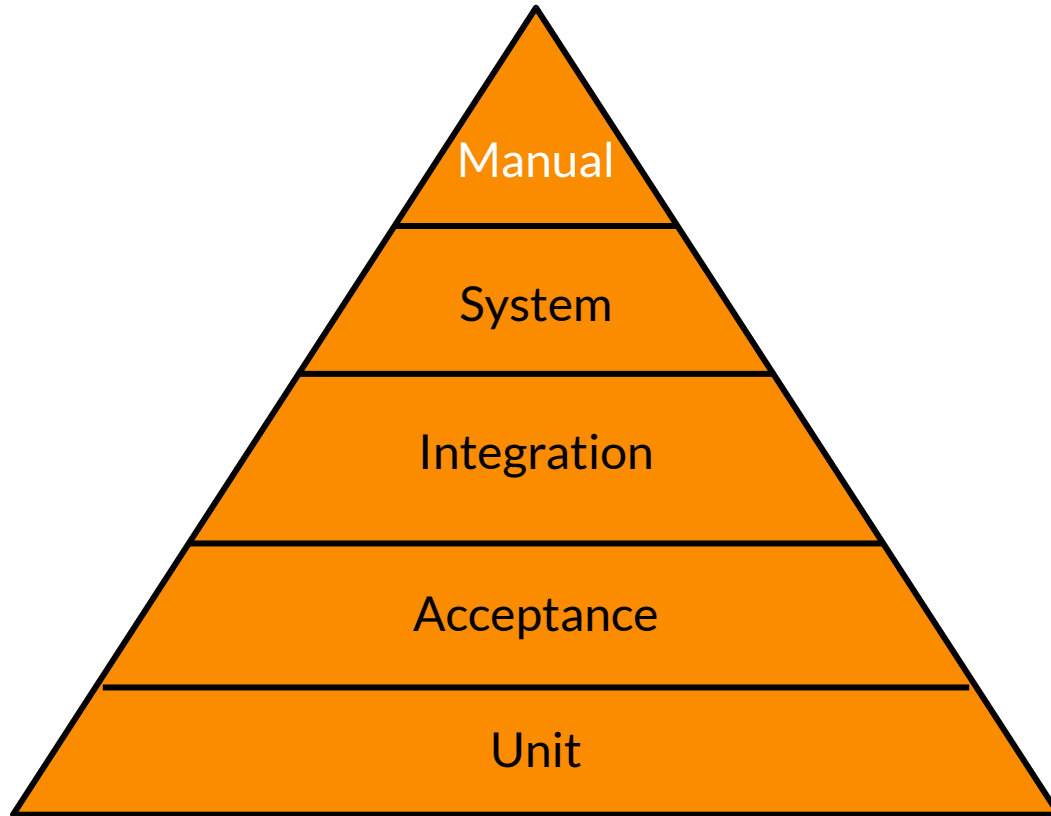
Uses all really services (Database, Email, ...)

May interact with many different parts of you app in a single test

May require some seeding of external services

5% Code Coverage

Manual Tests



Arrange - Act - Assert

```
public function testIsGreaterThan()
{
    // Arrange
    $five = new Integer(5);
    $four = new Integer(4);

    // Act
    $isGreaterThan = $five->isGreaterThan($four);
    $notGreaterThan = $four->isGreaterThan($five);

    // Assert
    $this->assertTrue($isGreaterThan);
    $this->assertFalse($notGreaterThan);
}
```

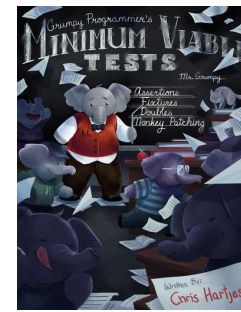
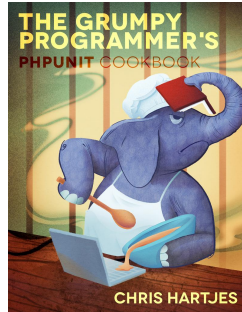
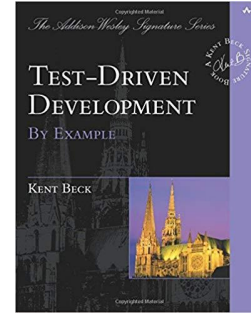
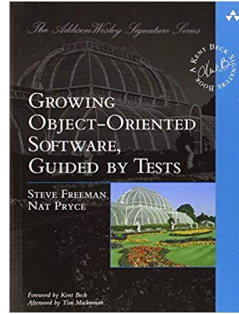
```
public function testIsGreaterThan()
{
    // Arrange
    $five = new Integer(5);
    $four = new Integer(4);

    // Act & Assert
    $this->assertTrue($five->isGreaterThan($four));
    $this->assertFalse($four->isGreaterThan($five));
}
```

Lets Mob Program

Practise Make Perfect

Resources



Questions

Feedback

Twitter: @braddle

Email: braddle@gmail.com

Thank you
